Proceedings of the 17th World Congress
The International Federation of Automatic Control
Seoul, Korea, July 6-11, 2008

A Modular Synthesis Approach for Distributed Safety Controllers, Part A:* Modelling and Specification

Dirk Missal and Hans-Michael Hanisch *

* *Martin Luther University Halle-Wittenberg,*
Institute for Computer Science
06099 Halle, Germany
(Dirk.Missal, Hans-Michael.Hanisch)@informatik.uni-halle.de

Abstract: The contribution provides an approach for formal synthesis of controllers ensuring safe operation on the shop floor level. It is structured into two parts.
Part A presents an introduction and a survey of related work. It gives a definition of a modular DEDS model that extends ordinary Petri nets in order to provide a modular, compositional approach for designing models of the uncontrolled plant behaviour and to define controllable local state transitions. Specifications are given in terms of safety properties by means of state predicates that may be local to each partial plant or even global ones spanning across more than one partial plant. The synthesized controllers have to prevent the specified states.
The major novel result of this part of the contribution is the definition of the behaviour of the plant without its complete composition. This means that the behaviour can be studied by means of modular steps within the modules and their interaction across module boundaries. This provides the basis for Part B of the contribution. An example taken from a physically real lab manufacturing system illustrates the modelling methodology and provides the plant model for the synthezis approach that is presented in Part B.

1. INTRODUCTION

Modern control systems in manufacturing are highly decentralized and constitute a network of locally and functionally distributed controllers that have to communicate to perform the control tasks that are given for specific manufacturing systems.

Theoretical approaches for the synthesis of controllers or supervisors have to be adopted to answer the newly emerging questions of distribution and communication. This includes topics like modular plant models that preserve information about the modular structure of the plant, methods to specify forbidden or desired behaviour locally instead of global specifications and synthesis approaches for local controllers and their communication instead of a global controller or supervisor.

Controller or supervisor synthesis has been established by the pioneering work of Ramadge and Wonham in the 80's of the last century and since then it has become a major field in research on Discrete Event Dynamic Systems (DEDS), mostly in academia.

In most cases, supervisors are synthesized on the process control level assuming that there are underlying controllers that run the process on the lower shop floor level. As a consequence the plant models for supervisor synthesis can abstract from many details and are therefore less complex and easier to design and to handle, since they are built from aggregated states and transitions. Things, however, look quite different if one goes down to the shop floor level and tries to apply the same methodologies there.

Models of uncontrolled plant behaviour turn out to be huge since they have to represent each detail of the physical equipment behaviour as well as the processed work piece, including actuators and sensors as well. Things are getting even worse if the goal is not to synthesize a single, monolithic controller but a collection of several, distributed controllers that have to interact with the plant as well as to communicate via a network to exchange information. Such a setup is exactly the field of research this work tries to contribute to.

This contribution is therefore organized as follows. We will refer to related work in Sec. 2 and point out the original approach we present. In Sec.3 the modular behaviour model is presented together with the modelling example used for the controller synthesis in Part B Missal and Hanisch (2008). The additionally needed formal specification is addressed in Sec. 3.3. The contribution is concluded by a summary in Sec. 4.

2. RELATED WORK

The approaches for control and supervisor synthesis differ in the used behaviour models as well as in the control structure. The area of supervisory control theory is based on finite state machine models. Modular algorithms were

* The work is supported by the Deutsche Forschungsgemeinschaft (DFG) under reference HA 1886/16-1 and HA 1886/16-2.

introduced early to reduce the complexity (Ramadge and Wonham (1986)). The complexity is generally a problem of algorithms, especially if an explicit state description is used. That holds also for the use of reachability analysis in implicit state description models which has to be avoided therefore (Krogh and Holloway (1991); Hanisch et al. (1996)). Hence, we combine the implicit system model of Net Condition/ Event systems with synthesis algorithms that avoid the computation of the state space.

There are different control structures discussed within the field of distributed control synthesis. There are distributed controllers without communication as in T.-S.Yoo and Lafortune (2002); Haji-Valizadeh and Loparo (1994) on one hand and with communication on the other (van Schuppen (1998); G.Barrett and Lafortune (1998); Rohloff and van Schuppen (2005); Iordache and Antsaklis (2006); Leduc and Dai (2007)). The distributed controllers with communication can interchange local observations or variables defining local states. The controllability is restricted by the use of distributed controllers without collaboration. We therefore focus on controllers with communication. More precisely we synthesize distributed controllers with communication of state information as discussed for example in Guan (2000). An alternative would be the communication of local observations between the controllers.

The presented approach for synthesis of distributed safety controllers with communication works on a modular plant model. A modular model semantics is applied as presented in Missal and Hanisch (2007). Control functions and communication variables and functions are obtained directly in contrast to the monolithic approach in Missal and Hanisch (2006). We can show a reduction of complexity in the calculation and for the synthesized functions.

3. MODELLING AND SPECIFICATION

The distributed control synthesis generally is based on formal models of the uncontrolled plant behaviour and systems specifications. The synthesis approach addressed in Part B is based on a modular *safe Net- Condition/Event system* ($_SNCES$) model and a specification in terms of predicates. The modelling with $_SNCES$ provides structural advantages during the modelling process as well as for the following synthesis. Manufacturing systems are mostly assembled of small units. The complexity of a system often only depends on the number of such elements. The modular structure of $_SNCES$ reflects the compound structure of systems. It offers a natural way to model a possibly large behaviour by combining models of every unit. Once models for plant elements are designed, they can be used over and over again. Many elements like a binary sensor type for example are used many times within the same plant.

3.1 The $_SNCES$ model

The safe Net- Condition/Event system is a special case of $NCES$.

Some elements of $_SNCES$ are common from Petri nets too. Within the model two kinds of modules are defined. The first kind are *basic modules*.

Definition 3.1. A safe Net- Condition/Event module ($_SNCEM$) is a tuple:

$$_SNCEM = \{P, T, F, CN, EN,$$
$$C^{in}, E^{in}, C^{out}, E^{out}, CI^{arc}, EI^{arc}, CO^{arc}, EO^{arc}, em, m_0\}$$

where: P is the set of places p, T is the set of transitions t, $F \subseteq (P \times T) \cup (T \times P)$ is the set of (ordinary) arcs, $CN \subseteq P \times T$ is the set of condition signals, $EN \subseteq T \times T$ is the set of event signals, C^{in} is the set of condition inputs c^{in}, E^{in} is the set of event inputs e^{in}, C^{out} is the set of condition outputs c^{out}, E^{out} is the set of event outputs e^{out}, $CI^{arc} \subseteq C^{in} \times T$ is the set of condition input arcs, $EI^{arc} \subseteq E^{in} \times T$ the set of event input arcs, $CO^{arc} \subseteq P \times C^{out}$ the set of condition output arcs, $EO^{arc} \subseteq T \times E^{out}$ the set of event output arcs, $em : T \rightarrow \{\boxbslash, \boxslash\}$ is the event mode for every transition,
EN is cycle free, i.e.:

(1) $\nexists (t_1, t_2) \in EN : t_1 = t_2$ and
(2) $\nexists ((t_1, t_2), \cdots, (t_{i-1}, t_i)) : (t_{l-1}, t_l) \in EN$ for $2 \leq l \leq i \wedge (t_1 = t_i)$

$m_0 : P \rightarrow \{0, 1\}$ is the initial marking.

□

The event mode for a transition declares whether incoming event signals at a transition are combined in "OR" or "AND" mode. The "AND" mode is default, and the corresponding symbols are omitted.

Every transition without an incoming event signal is called *trigger transition*. Transitions with one or more incoming event signals at a transition are called *forced transition*. Thus, the event mode is meaningful only for forced transitions.

The second kind of modules is defined for hierarchical combination of basic modules and is called *composite module*. Composite modules M_C consist of submodules $Sub(M_C)$ which can be basic modules or composite modules, their signal interconnections and a set of signal inputs and outputs as defined for basic modules. Composite modules are defined as follows:

Definition 3.2. A safe Net- Condition/Event system is inductively defined as follows:

(1) Every safe Net- Condition/Event module M_B is a safe Net- Condition/Event system.
(2) Every tuple $M_C = (Sub(M_C), \Phi, CK, EK)$ is a $_SNCES$ iff
   (a) $Sub(M_C) = (M_1, M_2, \ldots, M_K)$ is a finite, non empty set of safe Net- Condition/Event systems. Every $M_x \in Sub(M_C)$ is called *submodule* of M_C for which holds $M_C \notin Sub(M_C)$.
   (b) $\Phi = (C^{in}, E^{in}, C^{out}, E^{out})$ is an I/O set.
   (c)
$$CK \subseteq \bigcup_{i \in \{1,\ldots,k\}} (C^{in} \times C_i^{in}) \cup$$
$$\bigcup_{i,j \in \{1,\ldots,k\}} (C_i^{out} \times C_j^{in}) \cup \bigcup_{i \in \{1,\ldots,k\}} (C_i^{out} \times C^{out})$$
   describes the *condition interconnection* within M_C, for which furthermore

$\forall c_s \in (C^{out} \cup \bigcup_{i\in\{1,\ldots,k\}} C_i^{in}) : |\{c_q|(c_q, c_s) \in CK\}| \leq 1$ holds. [1]

(d)

$$EK \subseteq \bigcup_{i\in\{1,\ldots,k\}} (E^{in}\times E_i^{in}) \cup \bigcup_{i,j\in\{1,\ldots,k\}} (E_i^{out}\times E_j^{in}) \cup \bigcup_{i\in\{1,\ldots,k\}} (E_i^{out} \times E^{out})$$

describes the *event interconnection* within M_C. It is supposed that
$\forall e_s \in (E^{out} \cup \bigcup_{i\in\{1,\ldots,k\}} E_i^{in}) : |\{e_q|(e_q, e_s) \in EK\}| \leq 1$.

M_C is called a *composite module*. If all submodules of M_C are basic modules, M_C is called *basic system*.

□

The input state of ${}_SNCE$ modules is defined depending on steps ξ defined in Def. 3.7 as follows:

Definition 3.3. The input state *is* of a ${}_SNCEM$ is a mapping $is : C^{in} \cup E^{in} \to \{0, 1\}$ assigning a value of $\{0, 1\}$ to each signal input.
The value of signal inputs $c^{in} \in CK$ and $e^{in} \in EK$ is defined as follows:

$$c^{in} = \begin{cases} 1 & if\ \exists p \in P : m(p) = 1 \wedge (p, c^{out}) \in CO^{arc} \wedge \\ & \quad \exists ck \in CK : c^{out} \in ck \wedge c^{in} \in ck \\ 0 & else \end{cases}$$

$$e^{in} = \begin{cases} 1 & if\ \exists t \in \xi : (t, e^{out}) \in EO^{arc} \wedge \exists ek \in EK : \\ & \quad e^{out} \in ek \wedge e^{in} \in ek \\ 0 & else \end{cases}$$

□

For enabling of a transition and of a step, only the marking of places and the input state of a module are of interest.

Definition 3.4. A transition $t \in T$ of a ${}_SNCEM$ is:

(1) *marking enabled* at a marking m iff
$(\forall p \in P\ with\ (p, t) \in F : m(p) = 1) \wedge (\forall p \in P\ with\ (t, p) \in F : m(p) = 0)$
(2) *condition enabled* at a marking m and an input state *is* iff
$\forall p \in P\ with\ (p, t) \in CN : m(p) = 1$ and
$\forall c^{in} \in C^{in}\ with\ (c^{in}, t) \in CI^{arc} : is(c^{in}) = 1$.

□

A transition is marking enabled if all preplaces are marked and all postplaces are unmarked. A transition is condition enabled if all places that are connected via condition arcs are marked and all connected condition inputs have the value one.

Based on these terms, we can define *modular steps* in general and *enabled modular steps* under consideration of the event mode in particular.

Definition 3.5. Let M be a ${}_SNCEM$ with the marking m, $T_M \subseteq T$ the set of transition within M, the input state *is* and $\xi_M \subset T$ a nonempty set of transitions within a module M_n.
ξ_M is a *modular step* within the module M iff

(1) $|\xi_M \cap (T_M^t)| \geq 1 \dot{\vee} \xi_M \cap (T^s) = 1$,
while $T_M^t := \{t_M^t \in T_M | \exists e^{in} : (e^{in}, t) \in EI^{arc}\}$,
$T^s := t_M^s \in T_M | \nexists t' \in T : (t', t) \in EN$ and
(2) for every transition $t \in \xi_M$ holds:
- $em(t) = \boxtimes \wedge ((\exists t' \in \xi_M : (t', t) \in EN) \vee (\exists e^{in} \in E^{in}\ with\ (e^{in}, t) \in EI^{in} : is(e^{in}) = 1))$ or
- $em(t) = \boxtimes \wedge ((\forall t'\ with\ (t', t) \in EN : t' \in \xi_M) \wedge (\forall e^{in} \in E^{in}\ with\ (e^{in}, t) \in EI^{in} : is(e^{in}) = 1))$ and

(3) all transitions are free of conflicts to each other.

Ξ_M is the set of steps within M.

ξ_M is called *enabled modular step* under m and *is* iff ξ_M is marking and condition enabled under m and *is* and there is no set of transitions with $\xi'_M = \xi_M \cup \{t\}$ within M, which is also a step and marking and condition enabled under m and *is*.

□

Transitions t^t are called *local trigger transitions*. For local trigger transitions the property of being *event input enabled* is defined.

Definition 3.6. A local trigger transition is event input enabled if condition 2) of Def.3.5 is satisfied.

□

The property has to be analysed under consideration of the definition of the input state of modules.

Next we define steps within a modular ${}_SNCES$ as sets of local steps.

Definition 3.7. Let N be a ${}_SNCES$ with the marking m, the input state *is* and $\xi \in T$ a nonempty set of transitions within N.
A *step* ξ within N is defined as follows:

(1) A local step ξ is a step ξ if $|\xi \cap (T^s)| = 1$, holds.
(2) Every union of local steps

$$\xi = \bigcup_{M_n \in N} \xi_M$$

is a step if
- $|\xi \cap (T^s)| = 1$ holds and
- for the local steps ξ_M holds: $\exists \{ek\} \in EK : \forall (t, e^{out}) \in EO^{arc} \wedge e^{out} \in \{ek\} | t \in \xi$ which *event input enables* all $t_M^t \in \xi_M$.

Ξ is the set of steps within N.

ξ is called *enabled step* under m and *is* iff all $\xi_M \subseteq \xi$ are enabled under m and *is* and there is no set of local steps with $\xi' = \xi \cup \{\xi_M\}$, which is also enabled under m and *is*.

□

The defined enabled steps are always maximal steps and contain exactly one trigger transition. Conflict transitions must not be part of the same step.
The effect of firing an enabled step is defined as follows.

Definition 3.8. Let M be a ${}_SNCEM$ with the marking m (a state) and the input state *is*.
If ξ is an enabled step under m and *is*, then ξ is enabled to fire. The follower marking m' is determined for $p \in M$ to:

[1] ($C_x^{out} bzw. C_x^{in}$ represent the set of condition inputs and outputs of a submodule M_x.)

$$m'(p) = \begin{cases} 1 & if\ \exists t \in \xi : (t,p) \in F \\ 0 & if\ \exists t \in \xi : (p,t) \in F \\ m(p) & else. \end{cases}$$

The notation $m(M)\,[\xi\rangle\, m'(M)$ means that $m'(M)$ is the follower marking of $m(M)$ by firing the step ξ. □

We can also say, that Def. 3.8 holds for modular steps ξ_M if a step ξ with $\xi_M \subseteq \xi$ exists.

3.2 Well-structured modelling example

Detailed behaviour models of the uncontrolled plant are used for controller synthesis on the shop floor level. The plant behaviour modelling is discussed more detailed in Missal and Hanisch (2007). In this section we give a short overview on modelling and present the example that is used for controller synthesis in Part B.

We model the behaviour of every physical element as for example sensors, actuators (valves, electrical relays), cylinders, drives and so on. The basic behaviour model of every element is encapsulated with in a basic module. Work piece properties are also modelled in basic modules. Such properties are physical properties of the process work pieces and position information belonging to them. These basic modules are composed in different hierarchy levels and form units. Our modelling example is a cut-out of a modular production system displayed in Fig. 1. It is further used for description of the modular control synthesis in Part B. A detailed description of the modular production system can be found on our website (testbed for distributed control) as well as the $_SNCES$ model of the whole testing station (website: testing station and Missal and Hanisch (2007)).

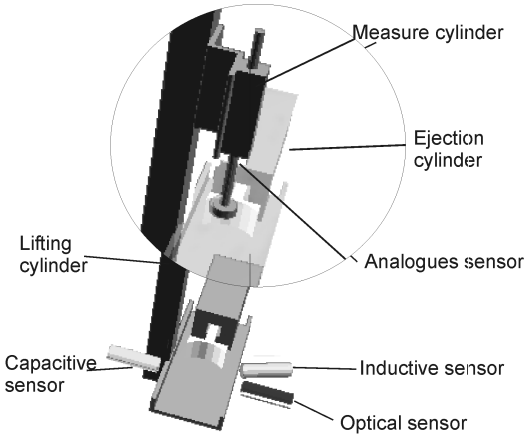

Fig. 1. Cutout of the testing station consisting of ejection module and measuring module.

At the testing station workpiece properties are checked. They are passed to further processing or rejected as scrap based on the property checks. The station is partitioned into four modules. The modules are the sensor model with sensors for the workpiece surface (colour, reflection), the lifting module for transportation of the pieces between a lower and an upper position, the ejection module for ejection of the workpieces at lower or upper position and the measure module measuring the height of the workpieces. The main elements of the testing station are labelled at Fig. 1. We choose the subsystems ejection module and measuring module for our example. The $_SNCES$ model of the ejection module and the measuring module and their interconnections are shown at Fig.2.

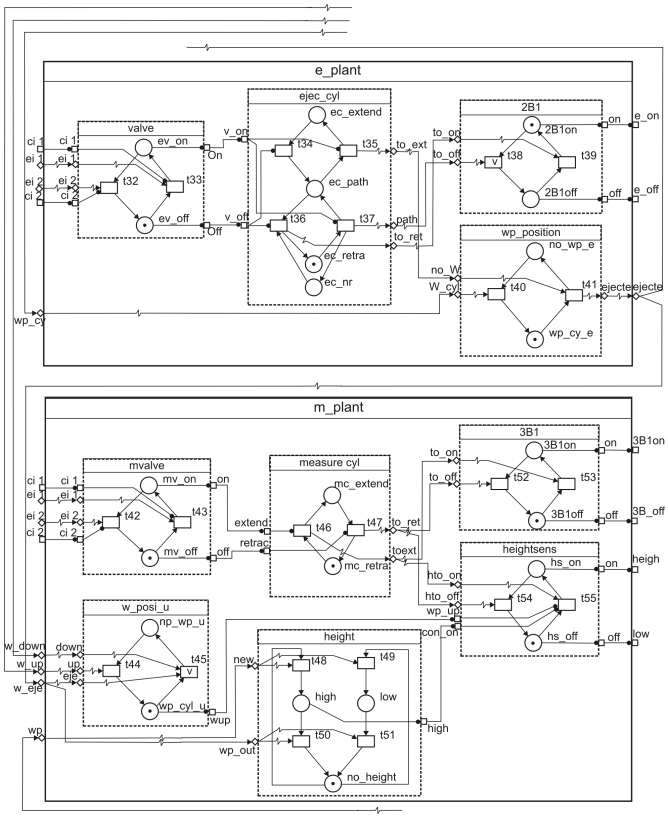

Fig. 2. Cutout of the sNCES model for the testing station

Some modelling rules (see Missal and Hanisch (2007)) are defined to support the efficient use of the models for our synthesis approach. Models following these rules are called well-structured. They are shortly introduced in the following.

The controllability of plant model elements is modelled by open signal inputs and their interconnections with transitions. Signal sink transitions of such interconnections are controllable. For the addressed control type of this paper we only consider condition inputs. The transitions $t1$ and $t2$ of the example model at Fig.2 are such controllable transitions.

Binary sensors are modelled by simple two state modules, just like relays for example. The sensors are connected to the observed plant element (e.g. cylinder) via event signals. That implies the assumption that sensors detect the plant behaviour surely. In contrast the interconnection of actuator modules and the influenced elements depend on the mode of functioning. The interconnection of magnetic valves and pneumatic cylinders is modelled by condition signals.

Beside the behaviour modelling, the structural (hierarchical) aspect is important for further use of the model. The

boundaries of the units of the plant naturally follow the composition of the physical elements of the real plant. The plant units should consist of the elements forming a more or less independent functional unit. Mostly actuators, moving elements and sensors form such units, as at the example of Fig.2. The structure of the model or its hierarchical composition respectively directly influences the modelling result. The presented approach synthesizes one distributed controller for every unit. The assignment of controllers to defined units is a decision of the human being who is responsible for structuring the plant model.

Within the unit models, all modules are completely composed as described in Thieme (2002). Through composition the signal interconnections between composed modules are transformed to signal arcs between the net elements and the module boundaries (dotted lines at Fig. 2) are dissolved. The unit modules themselves are not composed with other unit models. We finally get one basic module for every unit. The basic modules can have unplugged inputs and outputs as well as signal interconnections among each other. The modular synthesis approach at Part B is based on such modular model structure. The adequate composed example model is shown at Fig. 2 at Part B.

3.3 Specification

For formal synthesis of controllers we additionally need formal specifications of the required of forbidden system behaviour. For the synthesis of safety controllers we define a set of forbidden states in terms of predicates. The aim of constructing a safety controller is to prevent the system from reaching states which impose a risk to human, the system or the environment. Dangerous or forbidden states of the physical system are therefore formulated by predicates over the states of the model. The states of the model are represented by the marking of places. The state attributes of a forbidden state are defined as follows:

Definition 3.9. Let N be a ${}_SNCES$, $p \in P$ be a place of N and m a marking of N. A state atom ZA of N at m and p is a declaration:

$$ZA = [m(p) = a]; a \in \{0; 1\}.$$

A state predicate ZP of N on m is a function of state atoms:

$$ZP = ZA_1 \wedge ZA_2 \wedge \cdots \wedge ZA_n.$$

A state attribute ZE of N on m is a function of state predicates:

$$ZE = ZP_1 \vee ZP_2 \vee \cdots \vee ZP_n.$$

□

For every specified forbidden state a state attribute is defined. For the modular synthesis we use local specification predicates in the algorithm. The global specifications are decomposed into a conjunction of local state predicates. That is similar to the decomposition of supervisors in a conjunction of modular supervisors used and proofen in Ramadge and Wonham (1986).

The definition for the distribution of predicate functions is used for the distribution to local predicates as in Missal and Hanisch (2006) described for the distribution of control functions.

Definition 3.10. Let N be a ${}_SNCES$ and ZA state atoms and ZP state predicates, then local state atoms are defined as: $ZA_{M_x} \subseteq ZA : \forall p \in ZA_{M_x} | p \in M_x$.
The local state predicates ZP_{M_x} are:
$ZP_{M_x} \subseteq ZP : ZP_{M_x} = ZA_{M_x1} \wedge ZA_{M_x2} \wedge \cdots \wedge ZA_{M_xn}$,
while the following holds: $\forall ZA \in ZP : \exists ZP_{M_x}$ with $ZA \in ZP_{M_x}$ □

The local state predicates have to be linked with communication variables representing the association to a global function. Every local predicate of one global specification predicate has to be related to leastwise one other by a pair of communication variables $com_i^{(+,-)}$ (for both directions).

The distribution of the specification in terms of state attributes to local state predicates has to be performed at the first step of the synthesis algorithm.

In the following an example for specification of a forbidden state and its distribution is given. That specification is further used within the synthesis example at Part B of this contribution.

The forbidden state defines that measuring cylinder and ejection cylinder are not allowed to move simultaneously (see the model in Fig. 2). The specification example is just like in the example for the monolithic approach in Missal and Hanisch (2006). The analyzed state predicate following is:

$$p(ec\_nr) \wedge p(mc\_nr).$$

We have to transform the forbidden state predicate. The two state atoms are related to different modules and form local predicates. Because they have to represent a global state together we link them by communication variables. The communication variables $com_1^+; com_1^-$ are accociated to each local state predicate:

$$\{p(ec\_nr); com_1^+\}; \{p(mc\_nr); com_1^-\}.$$

4. SUMMARY

We have presented a modular and compositional model for modelling uncontrolled plant behaviour *without* complete composition as it was the case in previous work. A modular model semantics has been developed for this purpose.

We have further introduced state predicates for specification of forbidden plant behaviour and have shown how global predicates can be transformed into a set of local predicates and associated communications variables. Hence, we have the prerequisites for modular controller synthesis that is the subject of Part B.

REFERENCES

G.Barrett and S. Lafortune. On the synthesis of communicating controllers with decentralized informationstructures for discrete-event systems. In *Proceedings of the 37th IEEE Conference on Decision & Control, Tampa, FloridaUSA*. IEEE, December 1998.

X. Guan. *Distributed Supervisory Control of Forbidden Conditions, and Automated Synthesisand Composition of Task Controllers*. PhD thesis, The Graduate School University of Kentucky, 2000.

A. Haji-Valizadeh and K. A. Loparo. Decentralized supervisory predicate control of discrete event dynamicalsystems. In *Proceedings of the American Control Conference*, pages 1099–1103, June 1994.

H.-M. Hanisch, A. Lüder, and M. Rausch. Controller synthesis for net condition/event systems with incomplete stateobservation. In *Computer Integrated Manufacturing and Automation Technology (CIMAT 96)*, pages 351–356, Mai 1996.

M.V. Iordache and P.J. Antsaklis. *Supervisory Control of Concurrent Systems*. Systems and Control: Foundations & Applications. Birkhäuser Boston, 2006.

Bruce H. Krogh and Lawrence E. Holloway. Synthesis of feedback control logic for discrete manufacturing systems. *Automatica*, 27(4):641–651, 1991.

R.J. Leduc and P. Dai. Synthesis method for hierarchical interface-based supervisory control. In *Proc. of 26th American Control Conference*, pages 4260–4267, New York City, USA, July 2007.

D. Missal and H.-M. Hanisch. Synthesis of distributed controllers by means of a monolithic approach. In *Proceedings of the 11th IEEE Int. Conf. on Emerging Technologies and Factory Automation (ETFA'2006)*, pages 356–363, September 2006.

D. Missal and H.-M. Hanisch. Modular plant modelling for distributed control. In *IEEE Conference on Systems, Man, and Cybernetics*, pages 3475–3480, October 2007.

D Missal and H.-M. Hanisch. A modular synthesis approach for distributed safety controllers, part b: Modular control synthesis. In *Proceedings of the 17th IFAC World Congress*, Seoul, Korea, 2008.

P. J. Ramadge and W. M. Wonham. Modular supervisory control for discrete event systems. In *Seventh Internat. Conf. Analysis and Optimazition of Systems, Nice, France*, June 1986.

K.R Rohloff and J.H. van Schuppen. Approximation minimal communicated event sets for decentralized supervisorycontrol. In *Proceedings of the IFAC World Congress*, July 2005.

T.-S.Yoo and S. Lafortune. A general architecture for decentralized supervisory control of discrete-eventsystems. In *Journal of Discrete Event Dynamical Systems: Theory and Applications*, pages 335–377. 2002.

Modular testbed for distributed control. http://aut.informatik.uni-halle.de/forschung/testbed/.

Example testing station. *http://aut.informatik.uni-halle.de/forschung/synthese/example/*.

J. Thieme. *Symbolische Erreichbarkeitsanalyse und automatische Implementierung strukturierter,zeitbewerterter Steuerungsmodelle*. Hallenser Schriften zur Automatisierungtechnik. Logos-Verl., 2002.

J. H. van Schuppen. Decentralized supervisory control with information structures. In *Proceedings International Workshop on Discrete Event Systems (WODES98)*, pages 278–283. IEE Press, London, 1998.